

## **EXHIBIT B**

### **Configuring Inheritance for a Parent (Root) Descriptor**

**Oracle TopLink Developer's Guide 10g Release 3 (10.1.3)**

[http://www.oracle.com/technology/products/ias/toplink/doc/1013/main/\\_html/descfg020.htm](http://www.oracle.com/technology/products/ias/toplink/doc/1013/main/_html/descfg020.htm)

Copyright 1997, 2006, Oracle. All rights reserved.

[Skip Headers](#)

**Oracle TopLink Developer's Guide**  
**10g Release 3 (10.1.3)**  
 B13593-01


[Library](#) [Product](#) [Contents](#) [Index](#)

[Previous](#)[Next](#)

The script content on this page is for navigation purposes only and does not alter the content in any way.

## Configuring Inheritance for a Parent (Root) Descriptor

Inheritance describes how a derived (child) class inherits the characteristics of its superclass (parent). When you designate a class as a parent, you can configure how TopLink handles the class's inheritance hierarchy.

Table 28-23 summarizes which descriptors support parent inheritance configuration.

**Table 28-21 Descriptor Support for Parent Inheritance Configuration**

Descriptor	Using TopLink Workbench	Using Java
Relational Descriptors		
Object-Relational Descriptors		
EIS Descriptors		
XML Descriptors		

For more information about configuring inheritance for a child (branch or leaf) class descriptor, see "[Configuring Inheritance for a Child \(Branch or Leaf\) Class Descriptor](#)".

For more information, see "[Descriptors and Inheritance](#)".

## Using TopLink Workbench

To create a root class for an inheritance, use this procedure.

1. In the **Navigator**, select the descriptor you wish to specify as the root.
2. Choose the **Inheritance** tab in the **Property** window.

If the **Inheritance** tab is not visible, right-click the descriptor and choose **Select Advanced Properties > Inheritance**.

3. Select the **Is Root Parent Descriptor** option to specify this descriptor is a root class.

**Figure 28-33 Configuring Inheritance for a Root Descriptor, Inheritance Tab**

The screenshot shows the 'Inheritance' tab of the 'Order' descriptor's property window. The 'Is Root Parent Descriptor' radio button is selected. Under 'Field Selection', 'Specify Field' is chosen with a text box containing '<none selected>' and a 'Browse...' button. Under 'Indicator Selection', 'Use Class Indicator Dictionary' is selected. The 'Indicator Type' is set to 'String (java.lang)'. A table lists the indicator dictionary entries, with one entry for 'Order (examples.eis.attunit...'.

Include	Class	Indicator Value
<input type="checkbox"/>	Order (examples.eis.attunit...	

Use this table to complete the following root descriptor field on the Inheritance tab:

Field	Description
<b>Is Root Parent Descriptor</b>	Select this option to specify this descriptor as the root (parent) of the inheritance hierarchy.
<b>Use Class Extraction Method</b>	Choose this option to specify a class indicator using a class extraction method, and select your static class extraction method from the list.  For more information, see " <a href="#">Using Class Extraction Methods</a> ".
<b>Use Class Indicator Field</b>	Choose this option to specify a class indicator using a class indicator field.  For more information, see " <a href="#">Using Class Indicator Fields</a> ".
<b>Field Selection</b>	Choose the field to use as the class indicator field.
<b>Use XML Schema "Type" Attribute</b> <sup>Foot 1</sup>	Select this option to use the type attribute specified in the XML schema for this descriptor's reference class.
<b>Specify Field</b>	For a relational descriptor, select the field of the database table associated with this descriptor (see " <a href="#">Configuring Associated Tables</a> ").  For an EIS root descriptor (using XML records) or an XML descriptor, click <b>Browse</b> to select an element attribute or text node.
<b>Indicator Selection</b>	Choose between using a class name as the class indicator field value or specifying specific class indicator field values for each (nonabstract) child class.
<b>Use Class Name as Indicator</b>	Choose this option to use class names as the class indicator field value.
<b>Use Class Indicator Dictionary</b>	Choose this option to specify specific class indicator field values for each (nonabstract) child class.  When you choose this option, you must specify the data type of the class indicator field and the specific

Field	Description
	class indicator field values for each (nonabstract) child class.
<b>Indicator Type</b>	<p>Select the data type from the list to specify the data type of the class indicator field.</p> <p>To specify the specific class indicator field values for each (nonabstract) child class, click <b>Edit</b> and enter the appropriate value for each child class.</p>

Footnote <sup>1</sup> EIS root (see "EIS Root Descriptors") or XML descriptors (see "XML Descriptors") only.

## Using Java

Create a descriptor amendment method ("Configuring Amendment Methods") to customize the root class descriptor's InheritancePolicy using InheritancePolicy methods setParentClass, setClassIndicatorFieldName, addClassIndicator, useClassNameAsIndicator and setClassExtractionMethodName, as required.

Example 28-13 shows amendment methods for the Person and Student classes where Student extends Person in a relational project. In this example, a class indicator field is used (see "Using Class Indicator Fields").

### **Example 28-11 Configuring Inheritance for a Relational Root Class**

```

...
public static void addToPersonDescriptor(Descriptor descriptor)
{
    descriptor.getInheritancePolicy().setClassIndicatorFieldName
("CLIENT_TYPE");
    descriptor.getInheritancePolicy().addClassIndicator("P");
}

public static void addToStudentDescriptor(Descriptor descriptor)
{
    descriptor.getInheritancePolicy().setParentClass(Person.class);
    descriptor.getInheritancePolicy().setClassIndicatorFieldName
("CLIENT_TYPE");
    descriptor.getInheritancePolicy().addClassIndicator("S");
}

```

...

If you are using a class-extraction method (see ["Using Class Extraction Methods"](#)), you may also need to use InheritancePolicy methods `setOnlyInstancesExpression` and `setWithAllSubclassesExpression` (see ["Configuring Inheritance Expressions for a Parent \(Root\) Class Descriptor"](#)).

[Example 28-13](#) shows amendment methods for the `Person` and `Student` classes where `Student` extends `Person` in an EIS project using XML records. In this example, a class indicator field is used (see ["Using Class Indicator Fields"](#)).

### ***Example 28-12 Configuring Inheritance for an EIS Root Class***

...

```
public static void addToPersonDescriptor(Descriptor descriptor)
{
    descriptor.getInheritancePolicy().setClassIndicatorField(
        new XMLField("@CLIENT_TYPE")
    );
    descriptor.getInheritancePolicy().addClassIndicator("P");
}
```

```
public static void addToStudentDescriptor(Descriptor descriptor)
{
    descriptor.getInheritancePolicy().setParentClass(Person.class);
    descriptor.getInheritancePolicy().setClassIndicatorField(
        new XMLField("@CLIENT_TYPE")
    );
    descriptor.getInheritancePolicy().addClassIndicator("S");
}
```

...